FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

20MCA134 ADVANCED DATABASE MANAGEMENT SYSTEM LAB

LABORATORY RECORD

Name: APARNA K NAIR

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 2 Batch: A Roll No: 35

University Registration Number: FIT21MCA-2035

JULY 2022

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

CERTIFICATE

This is to certify that this is a Bonafide record of the Practical work done by ***APARNA K NAIR (FIT21MCA-2035)*** *in the* ***20MCA134 ADVANCED DATABASE MANAGEMENT SYSTEM*** *Laboratory towards the partial fulfilment for the award of the Master Of Computer Applications during the academic year 2021-2022.*

Signature of Staff in Charge

Name:

Signature of H O D

Name: Dr.Deepa Mary Mathews

Date of University practical examination ……

Signature of
Internal Examiner

Signature of
External Examiner

INDEX

COURSE OUTCOME 1

Experiment No:1

Aim: Create a table EMPLOYEE with following schema: (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name, Job_id , Salary)

a) Add a new column; HIREDATE to the existing relation.

b) Change the datatype of JOB_ID from char to varchar2.

c) Change the name of column/field Emp_no to E_no.

d) Modify the column width of the job field of emp table

Table Creation

create table EMPLOYEE(Emp_no varchar(3),E_name varchar(10),E_address varchar(10),E_ph_no int,Dept_no varchar(3),Dept_name varchar(5),Job_id char,Salary int);

SQL>Table created.

SQL> DESC EMPLOYEE;

Name	Null?	Type
EMP_NO		VARCHAR2(3)
E_NAME		VARCHAR2(10)
E_ADDRESS		VARCHAR2(10)
E_PH_NO		NUMBER(38)
DEPT_NO		VARCHAR2(3)
DEPT_NAME		VARCHAR2(5)
JOB_ID		CHAR(1)
SALARY		NUMBER(38)

a) SQL>alter table EMPLOYEE add Hire_Date date;
SQL>Table altered.
SQL> DESC EMPLOYEE;

Name	Null?	Type

EMP_NO		VARCHAR2(3)
E_NAME		VARCHAR2(10)
E_ADDRESS		VARCHAR2(10)
E_PH_NO		NUMBER(38)
DEPT_NO		VARCHAR2(3)
DEPT_NAME		VARCHAR2(5)
JOB_ID		VARCHAR2(2)
SALARY		NUMBER(38)
HIRE_DATE		DATE

b) SQL> alter table EMPLOYEE modify(Job_id varchar(2));
SQL>Table altered.
SQL> DESC EMPLOYEE;

Name	Null?	Type
EMP_NO		VARCHAR2(3)
E_NAME		VARCHAR2(10)
E_ADDRESS		VARCHAR2(10)
E_PH_NO		NUMBER(38)
DEPT_NO		VARCHAR2(3)
DEPT_NAME		VARCHAR2(5)
JOB_ID		VARCHAR2(2)
SALARY		NUMBER(38)
HIRE_DATE		DATE

c) SQL>alter table EMPLOYEE rename column Emp_no to E_no;
SQL>Table altered.
SQL> desc EMPLOYEE;

Name	Null?	Type
E_NO		VARCHAR2(3)
E_NAME		VARCHAR2(10)
E_ADDRESS		VARCHAR2(10)
E_PH_NO		NUMBER(38)
DEPT_NO		VARCHAR2(3)
DEPT_NAME		VARCHAR2(5)
JOB_ID		VARCHAR2(2)
SALARY		NUMBER(38)
HIRE_DATE		DATE

d) SQL>alter table EMPLOYEE modify Job_id varchar(20);
SQL>Table altered.
SQL> DESC EMPLOYEE;

Name	Null?	Type
E_NO		VARCHAR2(3)
E_NAME		VARCHAR2(10)
E_ADDRESS		VARCHAR2(10)
E_PH_NO		NUMBER(38)
DEPT_NO		VARCHAR2(3)
DEPT_NAME		VARCHAR2(5)
JOB_ID		VARCHAR2(20)
SALARY		NUMBER(38)
HIRE_DATE		DATE

Experiment No: 2

Aim: Create a table EMPLOYEE with following schema:

(Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id , Salary)

Write SQL queries for following question:

a) Insert aleast 5 rows in the table.

b) Display all the information of EMP table.

c) Display the record of each employee who works in department D10.

d) Update the city of Emp_no-12 with the current city as Nagpur.

e) Display the details of Employee who works in department MECH.

f) Delete the email_id of employee James.

g) Display the complete record of employees working in SALES Department.

Table Creation

create table employee2(emp_no varchar(10),emp_name varchar(20),emp_address varchar(20),emp_ph_no int,dept_no varchar(10),dept_name varchar(20),job_id varchar(10),salary int,email varchar(20),city varchar(10));

SQL>desc employee2;

Name	Null?	Type
EMP_NO		VARCHAR2(10)
EMP_NAME		VARCHAR2(20)
EMP_ADDRESS		VARCHAR2(20)
EMP_PH_NO		NUMBER(3,8)
DEPT_NO		VARCHAR2(10)
DEPT_NAME		VARCHAR2(20)

JOB_ID	VARCHAR2(10)
SALARY	NUMBER(3,8)
EMAIL	VARCHAR2(20)
CITY	VARCHAR2(10)

a). insert into employee2 values('e1','Ann','Thattil',9747716021,'d3','Production','j3',30000, 'ann123@gmail.com','Hyderabad');

insert into employee2 values('e2','Angel','Valluppara',6282719784,'d10','Sales','j1',20000, 'angel567@gmail.com', 'Pune');

insert into employee2 values('e3','Mariya','Puliyelil',9745458458,'d5','Marketing','j5',35000, 'mariya@gmail.com','Hyderabad');

insert into employee2 values('e12','Anna','Vadakkethala',7025362256,'d10','Sales','j2',22000, 'anna@gmail.com','Pune');

insert into employee2 values('e8','Anju','Kalaparambath',9847740623,'d5','Mechanical','j4', 18000, 'anju78@gmail.com','Banglore');

insert into employee2 values('e5','James','Parayil',9947676027,'d3','Production','j3',32000, 'jamespa23@gmail.com','Pune');

b). SQL> select * from employee2;

EMP_NO	EMP_NAME	EMP_ADDRESS	EMP_PH_NO	DEPT_NO	DEPT_NAME	JOB_ID	SALARY	EMAIL	CITY
e1	Ann	Thattil	9747716021	d3	Production	j3	30000	ann123@gmail.com	Hyderabad
e2	Angel	Valluppara	6282719784	d10	Sales	j1	20000	angel567@gmail.com	Pune
e3	Mariya	Puliyelil	9745458458	d5	Marketing	j5	35000	mariya@gmail.com	Hyderabad
e12	Anna	Vadakkethala	7025362256	d10	Sales	j2	22000	anna@gmail.com	Pune
e8	Anju	Kalaparambath	9847740623	d5	Mechanical	j4	18000	anju78@gmail.com	Banglore
e5	James	Parayil	9947676027	d3	Production	j3	32000	jamespa23@gmail.com	Pune

6 rows selected.

c). SQL>select * from employee2 where dept_no='d10';

EMP_NO	EMP_NAME	EMP_ADDRESS	EMP_PH_NO	DEPT_NO	DEPT_NAME	JOB_ID	SALARY	EMAIL	CITY
e2	Angel	Valluppara	6282719784	d10	Sales	j1	20000	angel567@gmail.com	Pune
e12	Anna	Vadakkethala	7025362256	d10	Sales	j2	22000	anna@gmail.com	Pune

d). SQL>update employee2 set city='Nagpur' where emp_no='e12';

```
SQL> select * from employee2;

EMP_NO     EMP_NAME             EMP_ADDRESS          EMP_PH_NO  DEPT_NO    DEPT_NAME            JOB_ID     SALARY     EMAIL                CITY
---------- -------------------- -------------------- ---------- ---------- -------------------- ---------- ---------- -------------------- ----------
e1         Ann                  Thattil              9747716021 d3         Production           j3         30000      ann123@gmail.com     Hyderabad
e2         Angel                Valluppara           6282719784 d10        Sales                j1         20000      angel567@gmail.com   Pune
e3         Mariya               Puliyelil            9745458458 d5         Marketing            j5         35000      mariya@gmail.com     Hyderabad
e12        Anna                 Vadakkethala         7025362256 d10        Sales                j2         22000      anna@gmail.com       Nagpur
e8         Anju                 Kalaparambath        9847740623 d5         Mechanical           j4         18000      anju78@gmail.com     Banglore
e5         James                Parayil              9947676027 d3         Production           j3         32000      jamespa23@gmail.com  Pune

6 rows selected.
```

e). SQL>select * from employee2 where dept_name='Mechanical';

```
EMP_N      EMP_NAME             EMP_ADDRESS          EMP_PH_NO  DEPT_NO    DEPT_NAME            JOB_ID     SALARY     EMAIL                CITY
---------- -------------------- -------------------- ---------- ---------- -------------------- ---------- ---------- -------------------- ----------
e8         Anju                 Kalaparambath        9847740623 d5         Mechanical           j4         18000      anju78@gmail.com     Banglore
```

f). update employee2 set email='' where emp_name='James';

```
1 row updated.

SQL> select * from employee2 where emp_name='James';

EMP_NO     EMP_NAME             EMP_ADDRESS          EMP_PH_NO  DEPT_NO    DEPT_NAMEJ           JOB_ID     SALARY     EMAIL                CITY
---------- -------------------- -------------------- ---------- ---------- -------------------- ---------- ---------- -------------------- ----------
e5         James                Parayil              9947676027 d3         Production           j3         32000                           Pune
```

g). select * from employee2 where dept_name='Sales';

```
EMP_NO     EMP_NAME             EMP_ADDRESS          EMP_PH_NO  DEPT_NO    DEPT_NAME            JOB_ID     SALARY     EMAIL                CITY
---------- -------------------- -------------------- ---------- ---------- -------------------- ---------- ---------- -------------------- ----------
e2         Angel                Valluppara           6282719784 d10        Sales                j1         20000      angel567@gmail.com   Pune
e12        Anna                 Vadakkethala         7025362256 d10        Sales                j2         22000      anna@gmail.com       Nagpur
```

Experiment No:3

Aim: Create a table EMPLOYEE with following schema: (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id, Designation , Salary)

Write SQL statements for the following query.

1. List the E_no, E_name, Salary of all employees working for MANAGER.

2. Display all the details of the employee whose salary is more than the Sal of any Manager

3. List the employees in the ascending order of Designations of those joined after 1981.

4. List the employees along with their Experience and Daily Salary.

5. List the employees who are Managers.

6. List the employees who joined on 1-MAY-81, 3-DEC-81, 17-DEC-81,19-JAN-80 .

7. List the employees who are working for the Deptno 10 or 3.

8. List the Enames those are starting with 'M' .

9. Dislay the name as well as the first five characters of name(s) starting with 'A'

10. List all the emps except 'Manager' in asc order of Salaries.

Table Creation

SQL> create table employee3(emp_no varchar(10),emp_name varchar(20),emp_address varchar(20),emp_ph_no int,dept_no varchar(10),dept_name varchar(20),job_id varchar(10),Designation varchar(10),salary int);

Table created.

insert into employee3 values('e1','Ann','Thattil',9747716021,'d3','Production','j3','Manager', 30000);

insert into employee3 values('e3','Anu','Parayil',9747452301,'d3','Production','j6','Supervisor', 18000);

insert into employee3 values('e3','Mariya','Puliyelil',9745458458,'d5','Marketing','j5', 'Manager',35000);

insert into employee3 values('e12','Anna','Vadakkethala',7025362256,'d10','Sales','j2', 'Manager',32000);

insert into employee3 values('e2','Angel','Valluppara',6282719784,'d10','Sales','j1', 'Supervisor',20000);

SQL> select * from employee3;

EMP_NO	EMP_NAME	EMP_ADDRESS	EMP_PH_NO	DEPT_NO	DEPT_NAME	JOB_ID	DESIGNATION	SALARY
e1	Ann	Thattil	9747716021	d3	Production	j3	Manager	30000
e3	Anu	Parayil	9747452301	d3	Production	j6	Supervisor	18000
e4	Mariya	Puliyelil	9745458458	d5	Marketing	j5	Manager	35000
e12	Anna	Vadakkethala	7025362256	d10	Sales	j2	Manager	32000
e2	Angel	Valluppara	6282719784	d10	Sales	j1	Supervisor	20000

1). SQL> select emp_no,emp_name,salary from employee3 where designation='Manager'

EMP_NO	EMP_NAME	SALARY
e1	Ann	30000
e3	Mariya	35000
e12	Anna	32000

2). SQL>select * from employee3 where salary > all (select salary from employee3 where dept_name='Production');

EMP_NO	EMP_NAME	EMP_ADDRESS	EMP_PH_NO	DEPT_NO	DEPT_NAME	JOB_ID	DESIGNATION	SALARY
e12	Anna	Vadakkethala	7025362256	d10	Sales	j2	Manager	32000
e4	Mariya	Puliyelil	9745458458	d5	Marketing	j5	Manager	35000

3). SQL>select * from employee3 where hire_date > '1-may-1981' order by designation asc;

EMP_NO	EMP_NAME	EMP_ADDRESS	EMP_PH_NO	DEPT_NO	DEPT_NAME	JOB_ID	DESIGNATION	SALARY	HIRE_DATE
e1	Ann	Thattil	9747716021	d3	Production	j3	Manager	30000	01-MAY-94
e2	Angel	Valluppara	6282719784	d10	Sales	j1	Supervisor	20000	03-DEC-81

4). SQL>select emp_no,emp_name,floor(months_between(current_date,hire_date)/12) as experience,(salary/30) as Daily_salary from employee3;

EMP_NO	EMP_NAME	EXPERIENCE	DAILY_SALARY
e1	Ann	27	1000
e3	Anu	42	600
e4	Mariya	42	1166.66667
e12	Anna	40	1066.66667
e2	Angel	40	666.666667

5). SQL> select * from employee3 where designation in('Manager');

EMP_NO	EMP_NAME	EMP_ADDRESS	EMP_PH_NO	DEPT_NO	DEPT_NAME	JOB_ID	DESIGNATION	SALARY	HIRE_DATE
e1	Ann	Thattil	9747716021	d3	Production	j3	Manager	30000	01-MAY-94
e4	Mariya	Puliyelil	9745458458	d5	Marketing	j5	Manager	35000	19-JAN-80
e12	Anna	Vadakkethala	7025362256	d10	Sales	j2	Manager	32000	01-MAY-81

6). SQL> select * from employee3 where hire_date in('1-MAY-81','3-DEC-81','17-DEC-81','19-JAN-80');

EMP_NO	EMP_NAME	EMP_ADDRESS	EMP_PH_NO	DEPT_NO	DEPT_NAME	JOB_ID	DESIGNATION	SALARY	HIRE_DATE
e3	Anu	Parayil	9747452301	d3	Production	j6	Supervisor	18000	19-JAN-80
e4	Mariya	Puliyelil	9745458458	d5	Marketing	j5	Manager	35000	19-JAN-80
e12	Anna	Vadakkethala	7025362256	d10	Sales	j2	Manager	32000	01-MAY-81

7). select * from employee3 where dept_no in('d10','d3');

EMP_NO	EMP_NAME	EMP_ADDRESS	EMP_PH_NO	DEPT_NO	DEPT_NAME	JOB_ID	DESIGNATION	SALARY	HIRE_DATE
e1	Ann	Thattil	9747716021	d3	Production	j3	Manager	30000	01-MAY-94
e3	Anu	Parayil	9747452301	d3	Production	j6	Supervisor	18000	19-JAN-80
e12	Anna	Vadakkethala	7025362256	d10	Sales	j2	Manager	32000	01-MAY-81
e2	Angel	Valluppara	6282719784	d10	Sales	j1	Supervisor	20000	03-DEC-81

8). select emp_no,emp_name from employee3 where emp_name like 'M%';

EMP_NO	EMP_NAME
e4	Mariya

9). select emp_no,emp_name from employee3 where length(emp_name)=5 and emp_name like 'A%';

EMP_NO	EMP_NAME
e2	Angel

10). select emp_no,emp_name,salary,designation from employee3 where designation not in('Manager') order by salary asc;

EMP_NO	EMP_NAME	SALARY	DESIGNATION
e3	Anu	18000	Supervisor
e2	Angel	20000	Supervisor

Experiment No: 4

Aim: Create the table described below.

Table Name : PRODUCT_MASTER

Description : used to store product information

Column name	Data type	size
PRODUCTNO	Varchar2	6
DESCRIPTION	Varchar2	15
PROFITPERCENT	Varchar2	4,2
UNITMEASURE	Varchar2	10
QTYONHAND	Number	8
REORDERLVL	Number	8
SELLPRICE	Number	8,2
COSTPRICE	Number	8,2

\

Table Name:CLIENT_MASTER

Description : used to store client information

Column name	Data type	size
CLIENTNO	Varchar2	6
NAME	Varchar2	20
ADDRESS1	Varchar2	30
ADDRESS2	Varchar2	30
CITY	Varchar2	15
PINCODE	Number	8
STATE	Varchar2	15
BALDUE	Number	10,2

Table Name : SALESMAN_MASTER

Description : used to store salesman information working for the company

Column name	Data type	size
SALESMANNO	Varchar2	6
SALESMANNAME	Varchar2	20
ADDRESS1	Varchar2	30
ADDRESS2	Varchar2	30
CITY	Varchar2	15
PINCODE	Number	8
STATE	Varchar2	15

generate SQL statements to perform the following computations on table data

a) list the names of all clients having 'a' as the second letter in their names.
b) listing of clients who stay in a city whose first letter is 'M'
c) list all clients who stay in 'Bangaluru' or 'Mangalore'
d) list all clients whose BalDue is greater than 10000
e) list products whose selling price is greater than 500 and less than or equal to 750
f) listing of names,city and state of clients who are not in the state of 'maharashtra'.
g) calculating the average price of all products.
h) determining the maximum and minimum price for the product prices.
i) count the number of products having the price greater than or equal to 500

Table Creation

create table product_master(prod_no varchar2(6),description varchar2(15),profit_percent number(4,2),unit_measure varchar2(10),qty_on_hand number(8),reorder_lvl number(8),sell_price number(8,2),cost_price number(8,2));

Table created.

SQL> create table client_master(client_no varchar2(6),name varchar2(20),address1 varchar2(30),address2 varchar2(30),city varchar2(15),pincode number(8),state varchar2(15),bal_due number(10,2));

Table created.

SQL> create table sales_master(salesman_no varchar2(6),salesman_name varchar2(20),address1 varchar2(30),address2 varchar2(30),city varchar2(15),pincode number(8),state varchar2(15));

Table created.

insert into product_master values('p1','avhsfhsv',20,150,55,12,590,500);

insert into product_master values('p2','jagfgfj',12,2000,100,700,520,650);

insert into product_master values('p3','dgjykkrk',8,100,60,400,800,1000);

```
select * from product_master;
```

PROD_N	DESCRIPTION	PROFIT_PERCENT	UNIT_MEASU	QTY_ON_HAND	REORDER_LVL	SELL_PRICE	COST_PRICE
p1	avhsfhsv	20	150	55	12	590	500
p2	jagfgfj	12	2000	100	700	520	650
p3	dgjykkrk	8	100	60	400	800	1000

insert into client_master values('c1','Ann','hgfheqf','gfgqfgqeg','Mangalore',680245, 'Karnataka',2000);

insert into client_master values('c2','Anju','sgfjgf','urtywhgb','Banglore',680278, 'Karnataka',40000);

insert into client_master values('c3','Nancy','parafhf','parafhf','Mumbai',545221, 'Maharashtra',20000);

insert into client_master values('c4','Sara','fhnfsfgrh','rhwhbdb','Kochi',680157,'Kerala', 5000);

```
select * from client_master;
```

CLIENT	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BAL_DUE
c1	Ann	hgfheqf	gfgqfgqeg	Mangalore	680245	Karnataka	2000
c2	Anju	sgfjgf	urtywhgb	Banglore	680278	Karnataka	40000
c3	Nancy	parafhf	parafhf	Mumbai	545221	Maharashtra	20000
c4	Sara	fhnfsfgrh	rhwhbdb	Kochi	680157	Kerala	5000

a). select client_no,name from client_master where name like '_a%';

CLIENT	NAME
c3	Nancy
c4	Sara

b). select client_no,name,city from client_master where city like 'M%';

CLIENT	NAME	CITY
c1	Ann	Mangalore
c3	Nancy	Mumbai

c). select client_no,name,city from client_master where city in('Mangalore','Banglore');

CLIENT	NAME	CITY
c1	Ann	Mangalore
c2	Anju	Banglore

d). select * from client_master where bal_due > 10000;

CLIENT	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BAL_DUE
c2	Anju	sgfjgf	urtywhgb	Banglore	680278	Karnataka	40000
c3	Nancy	parafhf	parafhf	Mumbai	545221	Maharashtra	20000

e). select * from product_master where sell_price > 500 and sell_price <= 750;

PROD_N	DESCRIPTION	PROFIT_PERCENT	UNIT_MEASU	QTY_ON_HAND	REORDER_LVL	SELL_PRICE	COST_PRICE
p1	avhsfhsv	20	150	55	12	590	500
p2	jagfgfj	12	2000	100	700	520	650

f). select name,city,state from client_master where state not in('Maharashtra');

NAME	CITY	STATE
Ann	Mangalore	Karnataka
Anju	Banglore	Karnataka
Sara	Kochi	Kerala

g). select avg(sell_price) as Average_price from product_master;

AVERAGE_PRICE
636.666667

h). select min(sell_price) as Min_price,max(sell_price) as Max_price from product_master;

MIN_PRICE	MAX_PRICE
520	800

i). select count(sell_price) as Count from product_master where sell_price > 500;

COUNT
3

Experiment No: 5

Aim:

Create **Sales table** with the following fields(Sales No, Salesname, Branch, Salesamount, DOB)

a) Insert five records

b) Calculate total salesamount in each branch

c) Calculate average salesamount in each branch .

Display the name and DOB of salesman in alphabetical order of the month.

Table Creation

create table sales(Sales_No varchar(10),Sales_name varchar(15),Branch varchar(15), Sales_amt number,DOB date);

a). insert into sales values('s1','Ann','Kochi',150000,'31-MAR-2001');

insert into sales values('s2','Angel','Thrissur',50000,'27-JUL-2000');

insert into sales values('s3','Mariya','Kochi',4000,'10-APR-2000');

insert into sales values('s4','Sree lakshmi','Thrissur',16000,'07-FEB-2000');

insert into sales values('s5','Krishna','Kozhikode',2000,'16-MAY-1999');

```
select * from sales;
```

SALES_NO	SALES_NAME	BRANCH	SALES_AMT	DOB
s1	Ann	Kochi	150000	31-MAR-01
s2	Angel	Thrissur	50000	27-JUL-00
s3	Mariya	Kochi	4000	10-APR-00
s4	Sree lakshmi	Thrissur	16000	07-FEB-00
s5	Krishna	Kozhikode	2000	16-MAY-99

b). select branch,sum(Sales_amt) as Total_sales from sales group by Branch;

BRANCH	TOTAL_SALES
Kozhikode	2000
Kochi	154000
Thrissur	66000

c). select branch,avg(Sales_amt) as Avg_sales from sales group by Branch;

BRANCH	AVG_SALES
Kozhikode	2000
Kochi	77000
Thrissur	33000

d). select Sales_name,to_char(DOB,'Month') as BornMonth from sales Order by to_Char (DOB,'Month');

SALES_NAME	BORNMONTH
Mariya	April
Sree lakshmi	February
Angel	July
Ann	March
Krishna	May

Experiment No: 6

Aim: Create an Emp table with the following fields:

(EmpNo, EmpName, Job,Basic, DA, HRA,PF, GrossPay, NetPay)

(Calculate DA as 30% of Basic and HRA as 40% of Basic)

a) Insert Five Records and calculate GrossPay and NetPay.
b) Display the employees whose Basic is lowest in each department .
c) If NetPay is less than Rs.10,000 add Rs.1200 as a special allowances
d) Display the employees whose GrossPay lies between 10,000 & 20,000

Table Creation

```
create table employee6(EmpNo varchar(10),EmpName varchar(20),Job varchar(15),
DeptName varchar(15),BasicPay number(8,2));

alter table employee6 add(DA number(8,2),HRA number(8,2),PF number(8,2));

a). insert into employee6(EmpNo,EmpName,Job,DeptName,BasicPay) values('e1','Ann',
'Manager','Sales',30000);

insert into employee6(EmpNo,EmpName,Job,DeptName,BasicPay) values('e2','Angel','
Supervisor','Sales',15000);

insert into employee6(EmpNo,EmpName,Job,DeptName,BasicPay) values('e3','Anna',
'Manager','Production',35000);

insert into employee6(EmpNo,EmpName,Job,DeptName,BasicPay) values('e4','Anju',
'Driver','Production',10000);

insert into employee6(EmpNo,EmpName,Job,DeptName,BasicPay) values('e5','Anil',
'Electrician','Production',22000);

update Employee6 set DA=BasicPay*.3;

update Employee6 set HRA=BasicPay*.4;

update Employee6 set PF=BasicPay*.12;

alter table employee6 add(GrossPay number(8,2),NetPay number(8,2));

update Employee6 set GrossPay=BasicPay+HRA+DA;

update Employee6 set NetPay=GrossPay-PF;
```

```
select * from employee6;
```

EMPNO	EMPNAME	JOB	DEPTNAME	BASICPAY	DA	HRA	PF	GROSSPAY	NETPAY
e1	Ann	Manager	Sales	30000	9000	12000	3600	51000	47400
e2	Angel	Supervisor	Sales	15000	4500	6000	1800	25500	23700
e3	Anna	Manager	Production	35000	10500	14000	4200	59500	55300
e4	Anju	Driver	Production	10000	3000	4000	1200	17000	15800
e5	Anil	Electrician	Production	22000	6600	8800	2640	37400	34760

b). select DeptName,min(BasicPay) as LowestBasicPay from employee6 group by DeptName;

DEPTNAME	LOWESTBASICPAY
Sales	15000
Production	10000

c). update employee6 set NetPay=NetPay+1200 where NetPay<20000;

SQL> select * from employee6;

EMPNO	EMPNAME	JOB	DEPTNAME	BASICPAY	DA	HRA	PF	GROSSPAY	NETPAY
e1	Ann	Manager	Sales	30000	9000	12000	3600	51000	47400
e2	Angel	Supervisor	Sales	15000	4500	6000	1800	25500	23700
e3	Anna	Manager	Production	35000	10500	14000	4200	59500	55300
e4	Anju	Driver	Production	10000	3000	4000	1200	17000	17000
e5	Anil	Electrician	Production	22000	6600	8800	2640	37400	34760

d). select EmpNo,EmpName,DeptName,GrossPay from employee6 where GrossPay between 10000 and 30000;

EMPNO	EMPNAME	DEPTNAME	GROSSPAY
e2	Angel	Sales	25500
e4	Anju	Production	17000

Experiment No: 7

Aim: Create a table called EMP with the following structure.

Name	Type
EMPNO	NUMBER (6)
ENAME	VARCHAR2 (20)
JOB	VARCHAR2 (10)
DEPTNO	NUMBER (3)
SAL	NUMBER (7,2)

a) Allow NULL for all columns except ename and job.
b) Add constraints to check, while entering the empno value (i.e) empno > 100.
c) Define the field DEPTNO as unique.
d) Create a primary key constraint for the table (EMPNO).
e) Write queries to implement and practice constraints

Table Creation

create table employee7(empno number(6),ename varchar(20) not null,job varchar(20) not null,deptno number(3),sal number(7,2));

Table created.

```
SQL> desc employee7;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                              NUMBER(6)
 ENAME                                     NOT NULL VARCHAR2(20)
 JOB                                       NOT NULL VARCHAR2(20)
 DEPTNO                                             NUMBER(3)
 SAL                                                NUMBER(7,2)
```

alter table employee7 modify empno check(empno>100);

Table altered.

```
SQL> desc employee7;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                              NUMBER(6)
 ENAME                                     NOT NULL VARCHAR2(20)
 JOB                                       NOT NULL VARCHAR2(20)
 DEPTNO                                             NUMBER(3)
 SAL                                                NUMBER(7,2)
```

```
alter table employee7 modify deptno unique;

Table altered.

alter table employee7 modify empno primary key;

Table altered.

insert into employee7 values(1200,'Abima','Operation Manager',23,12000);


insert into employee7 values(1200,'Abima','Operation Manager',23,12000);

insert into employee7 values(1200,'Abima','Operation Manager',23,12000)
*
ERROR at line 1:
ORA-00001: unique constraint (MCA09121.SYS_C0012049) violated
```

COURSE OUTCOME 2

Experiment No: 1

Aim: Write a pl/sql program to swap two numbers.

Source Code

```
declare
num1 number;
num2 number;
temp number;
begin
num1:=10;
num2:=20;
dbms_output.put_line('Before');
dbms_output.put_line('num1 = '|| num1 ||' num2 = '|| num2);
dbms_output.put_line('After');
temp:=num1;
num1:=num2;
num2:=temp;
dbms_output.put_line('num1 = '|| num1 ||' num2 = '|| num2);
end;
```

Output:

```
Statement processed.
Before
num1 = 10 num2 = 20
After
num1 = 20 num2 = 10
```

Experiment No: 2

Aim: Write a pl/sql program to find the largest of three numbers

Source code:

```
declare
num1 number;
num2 number;
num3 number;
begin
num1:=10;
num2:=20;
num3:=15;
dbms_output.put_line('num1 = '|| num1 ||' num2 = '|| num2 ||' num3 = '|| num3);
if num1>num2 and num1>num3
then
 dbms_output.put_line('Largest number is' || num1);
 else if num2>num1 and num2>num3
 then
 dbms_output.put_line('Largest number is' || num2);
 else
 dbms_output.put_line('Largest number is' || num3);
end if;
end;
```

Output:

```
Statement processed.
num 1=10 num 2=20 num3=15
Largest number is 20
```

Experiment No: 3

Aim: Write a pl/sql program to find the sum of digits in a given number

Source Code

```
declare
n integer;
temp_sum integer;
r integer;
begin
   n:=123456;
   temp_sum:=0;
   while n <>0
   loop
      r:=mod(n, 10);
      temp_sum := temp_sum + r;
      n:=Trunc(n / 10);
   end loop;
   dbms_output.Put_line('sum of digits = '|| temp_sum);
end;
```

Output

```
SQL> set serveroutput on
SQL> @3_3.sql
 16  /
sum of digits = 21
PL/SQL procedure successfully completed.
```

Experiment No: 4

Aim: Write a pl/sql program to display the number in reverse order.

Source code

```
declare
num number;
rev number;
begin
num:=&num;
rev:=0;
while num>0 loop
rev:=(rev*10) + mod(num,10);
num:=floor(num/10);
end loop;
dbms_output.put_line('Reverse of the number is: ' || rev);
end;
```

Output

```
@3_4.sql
13  /
Enter value for num: 3462
old   5: num:=&num;
new   5: num:=3462;
Reverse of the number is: 2643
PL/SQL procedure successfully completed.
```

Experiment No: 5

Aim: calculate the net salary and year salary if da is 30% of basic, hra is 10% of basic and pf is 7% if basic salary is less than 8000, pf is 10% if basic sal between 8000 to 160000.

Source code:

```
declare
   ename varchar2(15);
   basic number;
   da number;
   hra number;
   pf number;
   netsalary number;
begin
   ename:='&ename';
   basic:=&basic;
   da:=basic * (30/100);
   hra:=basic * (10/100);
      if (basic < 8000)
   then
      pf:=basic * (8/100);
      elsif (basic >= 8000 and basic <= 160000)
   then
      pf:=basic * (10/100);
end if;
      netsalary:=basic + da + hra -pf;
dbms_output.put_line('Employee name : ' || ename);
dbms_output.put_line('Providend Fund : ' || pf);
dbms_output.put_line('Net salary : ' || netsalary);
end;
```

Output

```
@3_5.sql
 27  /
Enter value for ename: ABC
old   9:     ename:='&ename';
new   9:     ename:='ABC';
Enter value for basic: 20000
old  10:     basic:=&basic;
new  10:     basic:=20000;
Employee name : ABC
Providend Fund : 2000
Net salary : 26000
PL/SQL procedure successfully completed.

SQL> @3_5.sql
 27  /
Enter value for ename: XYZ
old   9:     ename:='&ename';
new   9:     ename:='XYZ';
Enter value for basic: 5000
old  10:     basic:=&basic;
new  10:     basic:=5000;
Employee name : XYZ
Providend Fund : 400
Net salary : 6600
PL/SQL procedure successfully completed.
```

Experiment No: 6

Aim: write a PL/SQL code block that will accept an account number from the user, check if the users balance is less than minimum balance, only then deduct rs.100/- from the balance. This process is fired on the acct table.

Source code:

create table acct_master(acct_no number(5) primary key,acct_name varchar2(10),balance number(10));

insert into acct_master values(1, 'aaa', 1000);

insert into acct_master values(2, 'bbb', 100);

insert into acct_master values(3, 'ccc', 1100);

insert into acct_master values(4, 'ddd', 700);

insert into acct_master values(5, 'eee', 1700);

```
select * from acct_master;

ACCT_NO    ACCT_NAME    BALANCE
----------  ----------  ----------
1          aaa          1000
2          bbb          100
3          ccc          1100
4          ddd          700
5          eee          1700
```

```
declare

xacct_no number(5);

xmin_bal number(5):=1000;

xbalance number(5);

begin

xacct_no:=&xacct_no;

select balance into xbalance from acct_master where acct_no=xacct_no;

IF(xbalance < xmin_bal)

THEN

update acct_master set balance=balance-100 where acct_no=xacct_no;

xbalance:=xbalance-100;
```

```
dbms_output.put_line('Rs 100 is deducted and current balance is '||xbalance);
ELSE
dbms_output.put_line('Current balance is '||xbalance);
END IF;
END;
```

Output

```
@3_6.sql
 21  /
Enter value for xacct_no: 2
old   6: xacct_no:=&xacct_no;
new   6: xacct_no:=2;
Rs 100 is deducted and current balance is 0
PL/SQL procedure successfully completed.

@3_6.sql
 21  /
Enter value for xacct_no: 3
old   6: xacct_no:=&xacct_no;
new   6: xacct_no:=3;
Current balance is 1100
PL/SQL procedure successfully completed.
```

Experiment No: 7

Aim: Function that computes and returns the maximum of two values.

Source code:

```
DECLARE
   a number;
   b number;
   c number;
FUNCTION findMax(x IN number, y IN number)
RETURN number
IS
    z number;
BEGIN
   IF x > y THEN
      z:= x;
   ELSE
      Z:= y;
   END IF;
   RETURN z;
END;
BEGIN
   a:= 23;
    b:= 45;
   c := findMax(a, b);
   dbms_output.put_line(' Maximum of (23,45): ' || c);
END;
```

Output

```
@3_7.sql
 25  /
Maximum of (23,45): 45
```

Experiment No: 8

Aim: Function to check whether the string is palindrome or not.

Source code:

```
DECLARE
  s VARCHAR2(10) := 'abcbba';
  l VARCHAR2(20);
  t VARCHAR2(10);
BEGIN
  FOR i IN REVERSE 1..Length(s) LOOP
    l := Substr(s, i, 1);
    t := t
      ||''
      ||l;
  END LOOP;
  IF t = s THEN
  dbms_output.Put_line(t ||'' ||' is palindrome');
  ELSE
  dbms_output.Put_line(t ||'' ||' is not palindrome');
  END IF;
END;
```

Output

```
@3_8.sql
 19  /
abccba is palindrome
PL/SQL procedure successfully completed.
SQL> @3_8.sql
 19  /
abbcba is not palindrome
```

Experiment No: 9

Aim: To create and call a function that returns the total number of CUSTOMERS in the customers table.

Source code:

```
create table customer(cust_no number(5) primary key,cust_name varchar2(10),amount number(10));

insert into customer values(1, 'aaa', 1000);

insert into customer values(2, 'bbb', 100);

insert into customer values(3, 'ccc', 1100);

insert into customer values(4, 'ddd', 700);

insert into customer values(5, 'eee', 1700);
```

```
select * from customer;

CUST_NO    CUST_NAME    AMOUNT
----------  ----------  ----------
1           aaa         1000
2           bbb         100
3           ccc         1100
4           ddd         700
5           eee         1700
```

Function

```
CREATE OR REPLACE FUNCTION totalCustomers
RETURN number IS
  total number(2) := 0;
BEGIN
  SELECT count(*) into total FROM customer;
  RETURN total;
END;
/
```

Main

```
DECLARE
  c number(2);
BEGIN
```

```
  c := totalCustomers();
  dbms_output.put_line('Total no. of Customers: ' || c);
END;
/
```

Output

```
@3_9.sql
Function created.

SQL> @3_9_2.sql
Total no. of Customers: 5
PL/SQL procedure successfully completed.
```

Experiment No: 10

Aim: Write a procedure to Find the Sum of two numbers.

Source code:

```
create or replace procedure sum(n1 in int,n2 in int,result out int)
as
begin
result :=n1+n2;
end;

declare
result int;
begin
sum(5,5,result);
dbms_output.put_line(result);
end;
```

Output

Procedure created.

Statement processed.
10

Experiment No: 11

Aim: Create a Procedure that insert the rollno and name of a student into student table.

Source code:

create table Student28(roll_no varchar(5),name varchar(20));

Table created.

```
SQL> desc Student28;
 Name                 Null?    Type
 -------------------- -------- ----------------------------
 ROLL_NO                       VARCHAR2(5)
 NAME                          VARCHAR2(20)
```

create or replace procedure insertstud(rn in number,na varchar)

as

begin

insert into student values(rn,na);

end;

begin

insert into student values (28,'aaa');

insert into student values (12,'bbb');

insert into student values (21,'ccc');

dbms_output.put_line('Inserted Successfully');

end;

Output:

Statement processed.

Inserted Successfully

```
ROLL_NO      NAME
-------------- ----------
28           aaa
12           bbb
21           ccc
```

Experiment No: 12

Aim: Create a Procedure to get the count of instructors in the given department.

Table Creation:

```
create table inst(i_id integer ,i_name varchar(10),dept_id integer);


insert into inst values(100,'aaa',11);

insert into inst values(101,'bbb',12);

insert into inst values(102,'ccc',11);

insert into inst values(103,'ddd',13);

insert into inst values(104,'eee',11);
```

```
select * from inst;
I_ID   I_NAME      DEPT_ID
-----  ----------  ----------
100    aaa         11
101    bbb         12
102    ccc         11
103    ddd         13
104    eee         11
```

Source Code:

```
create or replace procedure icount
is
numm integer;
begin
dbms_output.put_line('The count is :');
select count(i_id) into numm from inst where dept_id=11;
dbms_output.put_line(numm);
end;
```

begin

icount;

end;

Out[ut:

```
Statement processed.
The count is :
3
```

Experiment No: 13

Aim: Use a cursor to display the details of customers.

Table Creation

create table customer(cust_id varchar(10),cust_name varchar(20),phone number);

insert into customer values('c1','aaa',8089264530);

insert into customer values('c2','bbb',8984236230);

insert into customer values('c3','ccc',9745852034);

insert into customer values('c4','ddd',8024879315);

insert into customer values('c5','eee',7569264538);

```
select * from customer;

CUST_ID      CUST_NAME          PHONE
----------   ----------         ----------
c1           aaa                8089264530
c2           bbb                8984236230
c3           ccc                9745852034
c4           ddd                8024879315
c5           eee                7569264538
```

Source Code:

```
DECLARE
   CURSOR cust_info IS
     SELECT cust_id,
          cust_name,
          phone
     FROM   customer;
   r_cust_info cust_info%ROWTYPE;
BEGIN
   OPEN cust_info;
   LOOP
      FETCH cust_info INTO r_cust_info;
      EXIT WHEN cust_info%NOTFOUND;
       dbms_output.Put_line('Customer Information:: '
```

```
                    ||'  ID: '
                    ||r_cust_info.cust_id
                    ||'  Name: '
                    ||r_cust_info.cust_name
                    ||'  Phone: '
                    ||r_cust_info.phone);
   END LOOP;
   dbms_output.Put_line('Total number of rows : '
                ||cust_info%rowcount);
   CLOSE cust_info;
END;
```

Output:

```
Statement processed.
Customer Information::  ID: c1  Name: aaa Phone: 8089264530
Customer Information::  ID: c2  Name: bbb Phone: 8984236230
Customer Information::  ID: c3  Name: ccc Phone: 9745852034
Customer Information::  ID: c4  Name: ddd Phone: 8024879315
Customer Information::  ID: c5  Name: eee Phone: 7569264538
Total number of rows : 5
```

Experiment No: 14

Aim: Use a cursor to display the details of employees of MCA department.

Table Creation:

create table employee(emp_id varchar(20),emp_name varchar(20),dept_id varchar(10), dept_name varchar(20));

insert into employee values('e1','aaa','d1','MCA');

insert into employee values('e2','bbb','d3','MBA');

insert into em;ployee values('e3','ccc','d1','MCA');

insert into employee values('e4','ddd','d2','CS');

insert into employee values('e1','eee','d3','MBA');

```
select * from employee;
EMP_ID     EMP_NAME          DEPT_ID    DEPT_NAME
---------- ----------------- ---------- ------------------
e1         aaa               d1         MCA
e2         bbb               d3         MBA
e3         ccc               d1         MCA
e4         ddd               d2         CS
e1         eee               d3         MBA
```

Source Code:

```
DECLARE
   CURSOR emp_info IS
     SELECT emp_id,
          emp_name,
          dept_id,
          dept_name
     FROM   employee WHERE dept_name='MCA';
   r_emp_info emp_info%ROWTYPE;
BEGIN
   OPEN emp_info;
   LOOP
      FETCH emp_info INTO r_emp_info;
```

```
    EXIT WHEN emp_info%NOTFOUND;
    dbms_output.Put_line('Customer Information:: '
                    ||'  ID: '
                    ||r_emp_info.emp_id
                    ||'  Name: '
                    ||r_emp_info.emp_name
                    ||' Dept_Id: '
                    ||r_emp_info.dept_id
                            ||' Dept_Name: '
                    ||r_emp_info.dept_name);
  END LOOP;
  dbms_output.Put_line('Total number of rows : '
                ||emp_info%rowcount);
  CLOSE emp_info;
END;
```

Output:

```
Statement processed.
Customer Information::   ID: e1  Name: aaa Dept_Id: d1 Dept_Name: MCA
Customer Information::   ID: e3  Name: ccc Dept_Id: d1 Dept_Name: MCA
Total number of rows : 2
```

Experiment No: 15

Aim: Create a row level trigger for the employee table that would fire for INSERT or UPDATE or DELETE operations performed on EMPLOYEE table. The trigger will display salary difference between old and new values.

Table Creation:

create table emp (eid number,ename varchar(20),sal numeric);

```
desc emp;
Column          Null?   Type
----------      ------  -------
EID             -       NUMBER
ENAME           -       VARCHAR2(20)
SAL             -       NUMBER
```

Source code:

```
CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON emp
FOR EACH ROW
DECLARE
  sal_diff number;
BEGIN
  sal_diff := :NEW.sal  - :OLD.sal;
  dbms_output.put_line('Old salary: ' || :OLD.sal);
  dbms_output.put_line('New salary: ' || :NEW.sal);
  dbms_output.put_line('Salary difference: ' || sal_diff);
END;
```

```
insert into emp values(12,'anju',20000);
```

```
1 row(s) inserted.
Old salary:
New salary: 20000
Salary difference:
```

```
update emp set sal=sal+1000 where eid=12;
```

```
1 row(s) updated.
Old salary: 20000
New salary: 21000
Salary difference: 1000
```

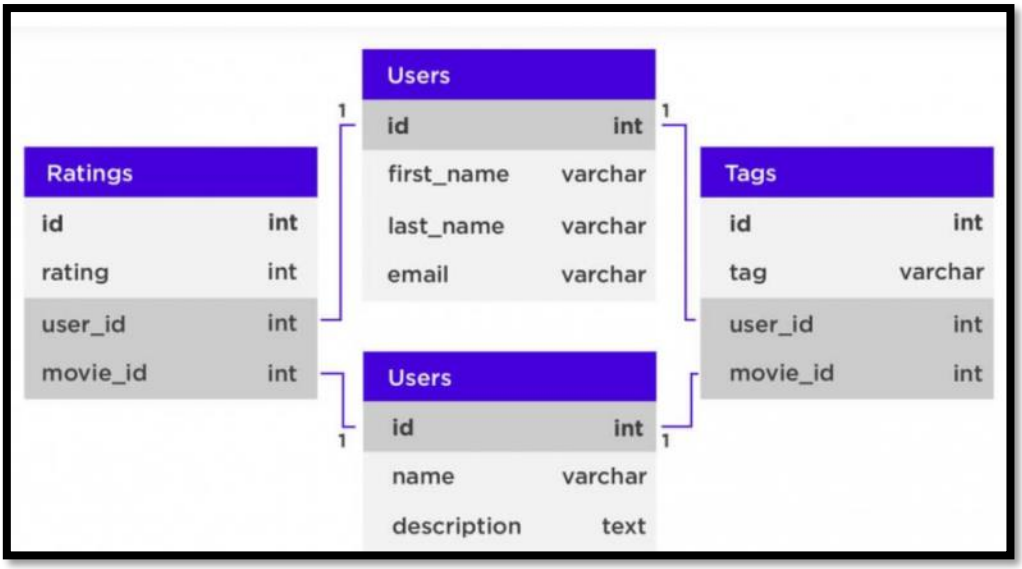
COURSE OUTCOME 3

1. Compare Relational & Non-Relational Database

A database is **a collection of information that is organized so** that it can be easily accessed, managed and updated. The two main types of databases are relational and non-relational. The main difference between these is how they store their information.

Relational Database

- A relational database is one that stores data in tables. Each column of the table represents the attribute or property of information that will be stored, and a row represents the value for that category.
- Each table can store data only for one object. To store the details about another object/entity, a new table is to be created. Then, connected tables form relationships. The relationship between tables and field types is called a **schema**. For relational databases, the schema must be clearly defined.
- Each table of the database has a specific key that identifies the data in the table. To connect one table to another, foreign keys are used.
- SQL is used to execute queries, retrieve data, and edit data by updating, deleting, or creating new records in the relational database
- Popular Relational/SQL Databases: SQL Server, MySQL, PostgreSQL



Non-relational Databases

- A **non-relational database** is any database that does not use the tabular schema of rows and columns like in relational databases. Rather, its storage model is optimized for the type of data it's storing.
- Non-relational databases are different from traditional relational databases in that they store their data in a non-tabular form.
- Non-relational databases are also known as **NoSQL databases** which stands for "Not Only SQL.
- There are four different types of NoSQL databases.
    1. **Document-oriented databases** –Document databases usually pair each key with a complex data structure (called a document).
    2. **Key-Value Stores** – This is a database that uses different keys where each key is associated with only one value in a collection.
    3. **Wide-Column Stores** uses tables, rows, and columns and names and format of the columns can vary from row to row in the same table.
    4. **Graph Stores** –uses graph structures for semantic queries with nodes, edges, and properties to represent and store data.

- Non-relational databases are often used when large quantities of complex data.
- Non-relational databases often perform faster
- Non-relational databases are ideal for storing data that may be changed frequently
- Popular Non-Relational/NoSQL Databases : MongoDB, Redis, Cassandra

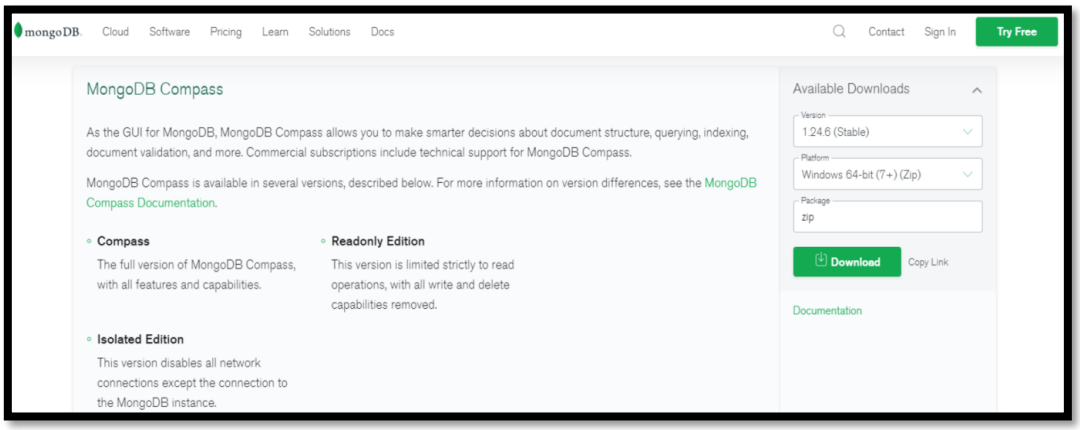
Key	Document
1001	{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }
1002	{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }

2. Installation of NoSQL Database – MongoDB

MongoDB, the most popular NoSQL database. The format of data storage in MongoDB is called BSON (similar to JSON format). **MongoDB Compass** is a graphical interface to interact with the MongoDB database management system.

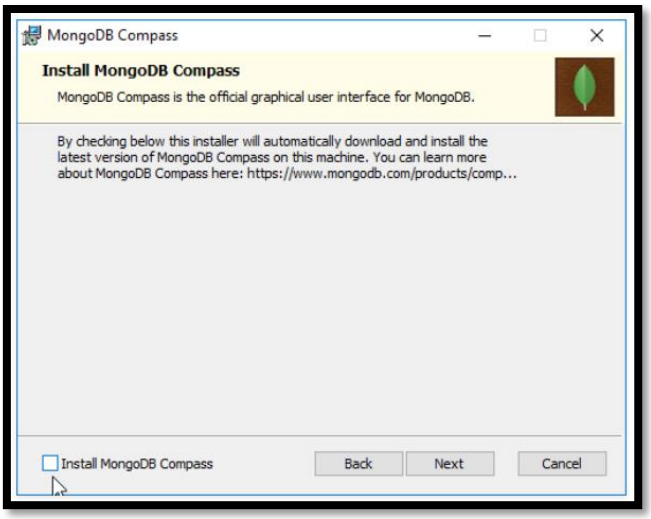
Steps for Installation:

1. Download MongoDB Compass from MongoDb website https://www.mongodb.com/try/download/compass

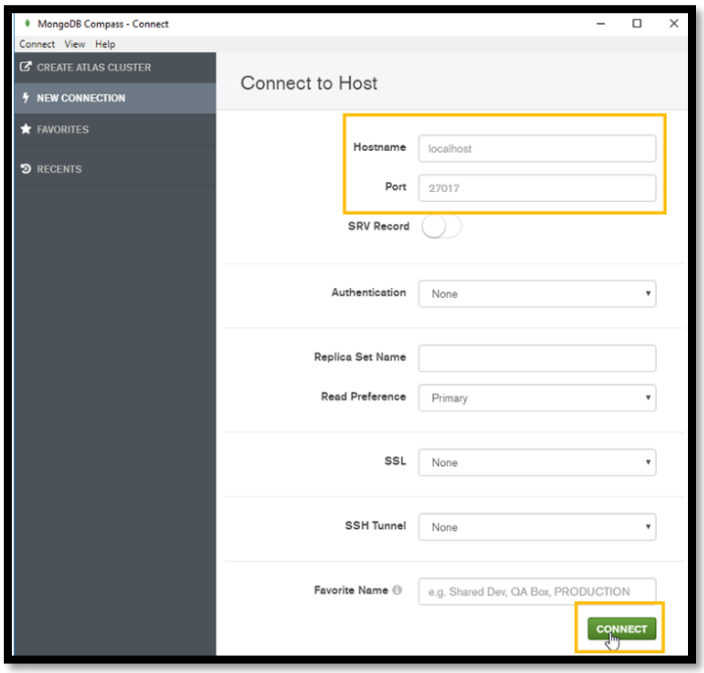


2. Unzip the downloaded File.
3. Double click the installer icon.

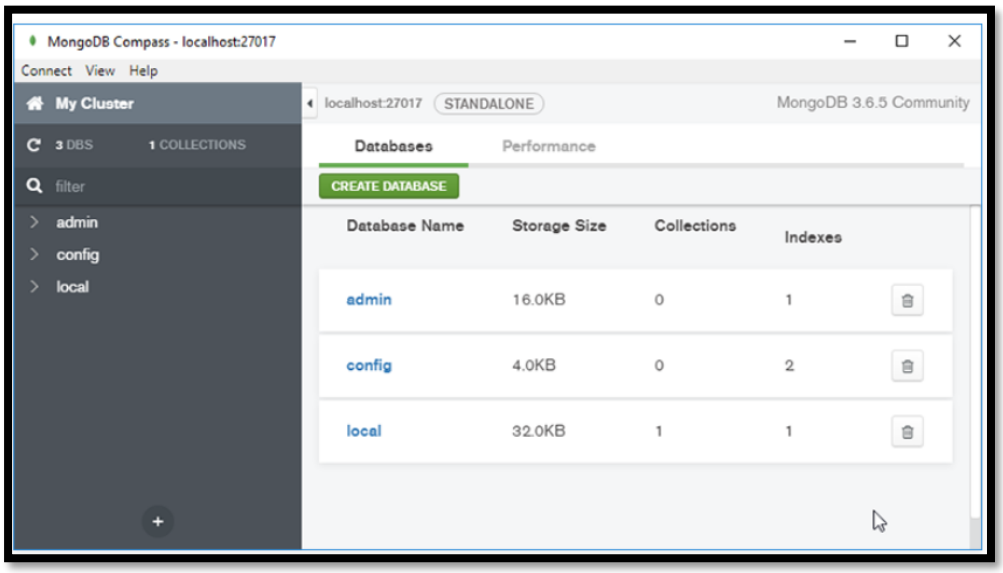
4. Follow the pop-ups to install MongoDB Compass GUI.



5. At this stage, a prompt will pop which can be used to configure the setting of the MongoDB Compass.



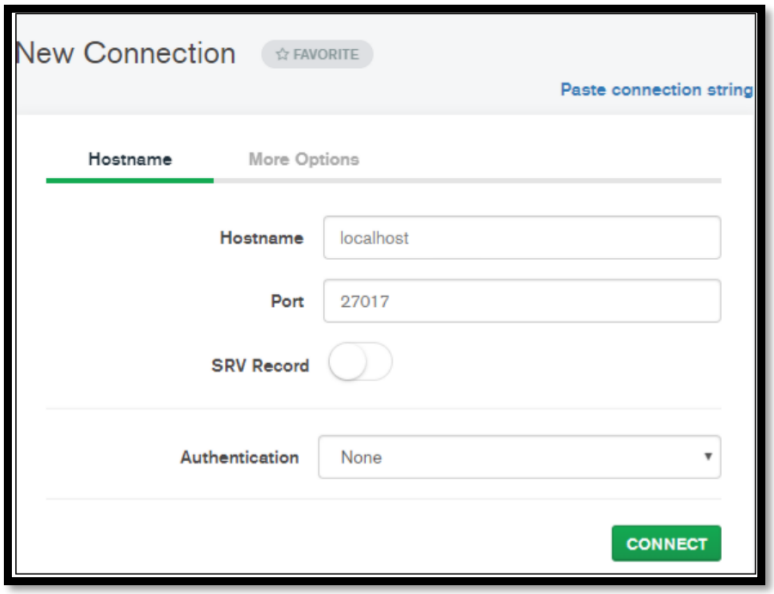
6. After the above step, all installation process is done and is ready to work with the databases.



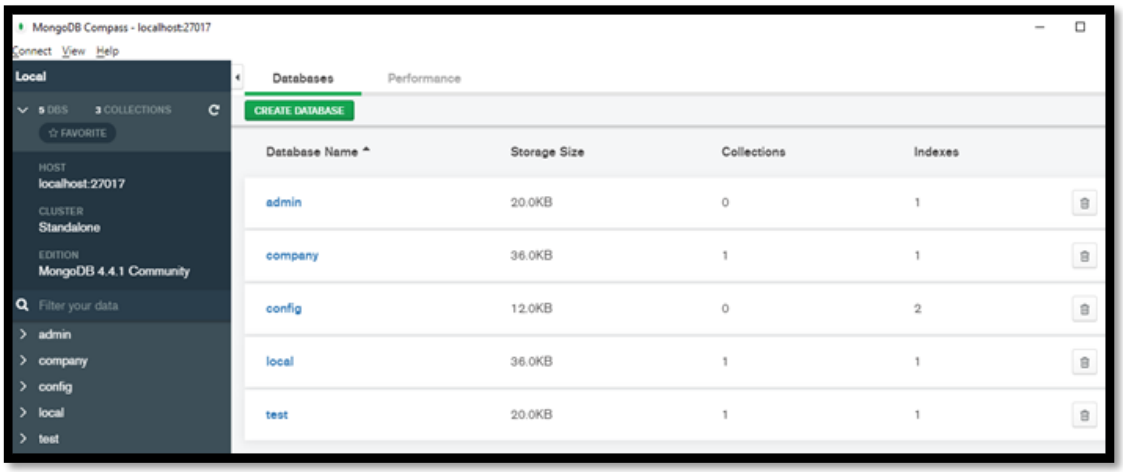
7. Establishing connection with MongoDB Compass.

   First, open the MongoDB Compass application and click the **Fill in connection fields individually** option. Specify the Hostname and the port in which your MongoDB server

is running. If you installed MongoDB on your machine with default settings, the Hostname would be the localhost, and the port is 27017. Then click **CONNECT**.

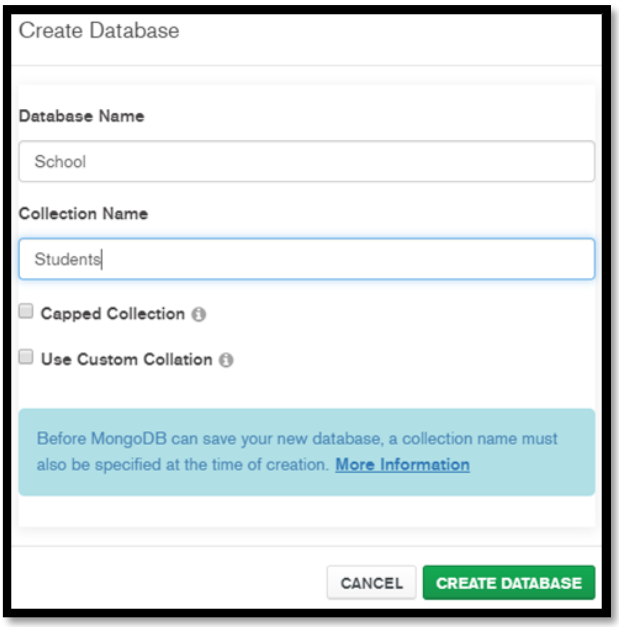


Now, you are connected to your MongoDB server. You can see a list of databases available in the server and a set of options that you can use to create and delete databases:

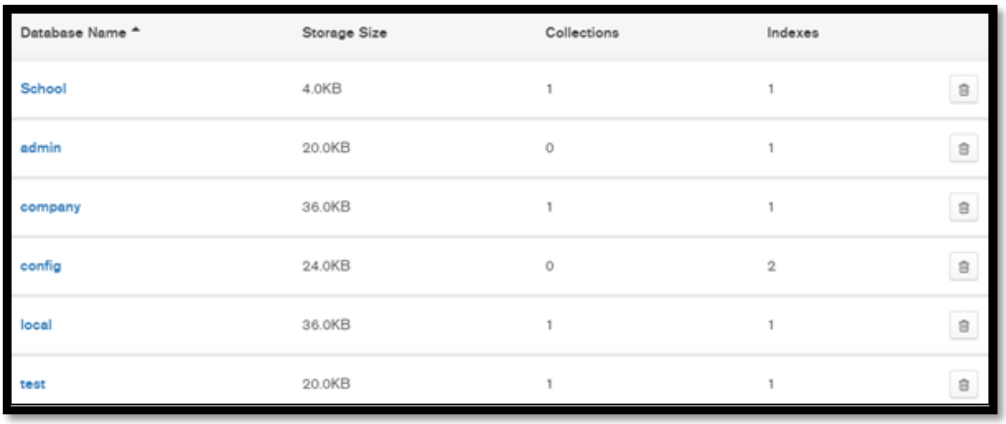


8. Creating databases, collections, and inserting data

   Click the **CREATE DATABASE** option to create a new database. A new window will pop up and enter the Database and Collection name , Click on the **CREATE DATABASE**



The newly created database will appear in the dashboard as



Database Name	Storage Size	Collections	Indexes
School	4.0KB	1	1
admin	20.0KB	0	1
company	36.0KB	1	1
config	24.0KB	0	2
local	36.0KB	1	1
test	20.0KB	1	1

COURSE OUTCOME 4

Experiment No: 1

Aim:

- Install the MongoDB Compass GUI and configure it.
- Create a collection student consists of details like rollno;,.', name, phoneno, marks, address, year of course etc Insert the details of the multiple students (atleast 5) in the form of documents in the student collection.
- Retrieve the fields rollno, name, phoneno, marks, city for all the documents in the collection student.
- Display the details of students who achieved a score more than 90 and are from 'Thrissur'.
- Update the phone number of Sujith in the student collection. Retrieve the updated information.
- Update the year of course in all the documents in the student collection to 2021. Also retrieve the updated information.
- Delete the details of the student whose name is 'Abhilash' from the student collection
- Retrieve the number of students per department from the student collection.
- Arrange the name of the students in ascending order along with all the columns.
- Rename *city* as *town* and add the detail of address consists of apartment no, street name and PIN.
- Display the contact address of 'Abhilash'.

Source code:

```
> db.createCollection("student");

{ "ok" : 1 }

> db.student.insertOne({ _id:1,Roll_no:101, name:"aaa", phone:576524687, marks:95, address:"abcd", city:"Thrissur",year:2020});

{ "acknowledged" : true, "insertedId" : 1 }

> db.student.insertOne({ _id:2,Roll_no:102, name:"bbb", phone:576524589, marks:85, address:"sdfg", city:"Kollam",year:2021});

{ "acknowledged" : true, "insertedId" : 2 }
```

```
> db.student.insertMany( [{_id:3, Roll_no:103, name:"ccc", phone:575127587, marks:75, address:"frhd",city:"Kannur", year:2021}, {_id:4,Roll_no:104, name:"ddd",phone:62827784, marks:99, address:"xyz",city:"Thrissur", year:2021},{_id:5,Roll_no:105, name:"eee",phone:415787784, marks:75, address:"prqs",city:"Ernakulam", year:2020}]);

{ "acknowledged" : true, "insertedIds" : [ 3, 4, 5 ] }

> db.student.find().pretty();

{
        "_id" : 1,
        "Roll_no" : 101,
        "name" : "aaa",
        "phone" : 576524687,
        "marks" : 95,
        "address" : "abcd",
        "city" : "Thrissur",
        "year" : 2020
}
{
        "_id" : 2,
        "Roll_no" : 102,
        "name" : "bbb",
        "phone" : 576524589,
        "marks" : 85,
        "address" : "sdfg",
        "city" : "Kollam",
        "year" : 2021
}
{
        "_id" : 3,
        "Roll_no" : 103,
        "name" : "ccc",
```

```
        "phone" : 575127587,
        "marks" : 75,
        "address" : "frhd",
        "city" : "Kannur",
        "year" : 2021
}
{
        "_id" : 4,
        "Roll_no" : 104,
        "name" : "ddd",
        "phone" : 62827784,
        "marks" : 99,
        "address" : "xyz",
        "city" : "Thrissur",
        "year" : 2021
}
{
        "_id" : 5,
        "Roll_no" : 105,
        "name" : "eee",
        "phone" : 415787784,
        "marks" : 75,
        "address" : "prqs",
        "city" : "Ernakulam",
        "year" : 2020
}

> db.student.find({},{Roll_no:1, name:1,phone:1, marks:1,city:1,_id:0});
{ "Roll_no" : 101, "name" : "aaa", "phone" : 576524687, "marks" : 95, "city" : "Thrissur" }
```

```
{ "Roll_no" : 102, "name" : "bbb", "phone" : 576524589, "marks" : 85, "city" : "Kollam" }

{ "Roll_no" : 103, "name" : "ccc", "phone" : 575127587, "marks" : 75, "city" : "Kannur" }

{ "Roll_no" : 104, "name" : "ddd", "phone" : 62827784, "marks" : 99, "city" : "Thrissur" }

{ "Roll_no" : 105, "name" : "eee", "phone" : 415787784, "marks" : 75, "city" : "Ernakulam" }


> db.student.find({$and:[{city:"Thrissur"} , {marks: {$gt:90}}]})

{ "_id" : 1, "Roll_no" : 101, "name" : "aaa", "phone" : 576524687, "marks" : 95, "address" :
"abcd", "city" : "Thrissur", "year" : 2020 }

{ "_id" : 4, "Roll_no" : 104, "name" : "ddd", "phone" : 62827784, "marks" : 99, "address" :
"xyz", "city" : "Thrissur", "year" : 2021 }


 db.student.update({name:"bbb"},{$set :{"phone" : 54789625}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.student.find({name:"bbb"},{Roll_no:1, name:1,phone:1, marks:1,city:1,_id:0})

{ "Roll_no" : 102, "name" : "bbb", "phone" : 54789625, "marks" : 85, "city" : "Kollam" }


> db.student.update({year:2020},{$set :{"year" : 2021}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.student.find({year:2021},{Roll_no:1, name:1,phone:1, marks:1,city:1,year:1,_id:0})

{ "Roll_no" : 101, "name" : "aaa", "phone" : 576524687, "marks" : 95, "city" : "Thrissur",
"year" : 2021 }

{ "Roll_no" : 102, "name" : "bbb", "phone" : 54789625, "marks" : 85, "city" : "Kollam",
"year" : 2021 }

{ "Roll_no" : 103, "name" : "ccc", "phone" : 575127587, "marks" : 75, "city" : "Kannur",
"year" : 2021 }

{ "Roll_no" : 104, "name" : "ddd", "phone" : 62827784, "marks" : 99, "city" : "Thrissur",
"year" : 2021 }

{ "Roll_no" : 105, "name" : "eee", "phone" : 415787784, "marks" : 75, "city" : "Ernakulam",
"year" : 2021 }


>db.student.deleteOne({name:"bbb"})

{ "acknowledged" : true, "deletedCount" : 1 }
```

```
> db.student.find({name:"bbb"},{Roll_no:1, name:1,phone:1, marks:1,city:1,year:1,_id:0})

> db.student.find()

{ "_id" : 1, "Roll_no" : 101, "name" : "aaa", "phone" : 576524687, "marks" : 95, "address" : "abcd", "city" : "Thrissur", "year" : 2021 }

{ "_id" : 3, "Roll_no" : 103, "name" : "ccc", "phone" : 575127587, "marks" : 75, "address" : "frhd", "city" : "Kannur", "year" : 2021 }

{ "_id" : 4, "Roll_no" : 104, "name" : "ddd", "phone" : 62827784, "marks" : 99, "address" : "xyz", "city" : "Thrissur", "year" : 2021 }

{ "_id" : 5, "Roll_no" : 105, "name" : "eee", "phone" : 415787784, "marks" : 75, "address" : "prqs", "city" : "Ernakulam", "year" : 2021 }


> db.student.find().sort({name:1})

{ "_id" : 1, "Roll_no" : 101, "name" : "aaa", "phone" : 576524687, "marks" : 95, "address" : "abcd", "city" : "Thrissur", "year" : 2021 }

{ "_id" : 3, "Roll_no" : 103, "name" : "ccc", "phone" : 575127587, "marks" : 75, "address" : "frhd", "city" : "Kannur", "year" : 2021 }

{ "_id" : 4, "Roll_no" : 104, "name" : "ddd", "phone" : 62827784, "marks" : 99, "address" : "xyz", "city" : "Thrissur", "year" : 2021 }

{ "_id" : 5, "Roll_no" : 105, "name" : "eee", "phone" : 415787784, "marks" : 75, "address" : "prqs", "city" : "Ernakulam", "year" : 2021 }


> db.student.updateMany( {}, { $rename: { "city": "town" } } )

{ "acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 4 }

> db.student.find()

{ "_id" : 1, "Roll_no" : 101, "name" : "aaa", "phone" : 576524687, "marks" : 95, "address" : "abcd", "year" : 2021, "town" : "Thrissur" }

{ "_id" : 3, "Roll_no" : 103, "name" : "ccc", "phone" : 575127587, "marks" : 75, "address" : "frhd", "year" : 2021, "town" : "Kannur" }

{ "_id" : 4, "Roll_no" : 104, "name" : "ddd", "phone" : 62827784, "marks" : 99, "address" : "xyz", "year" : 2021, "town" : "Thrissur" }

{ "_id" : 5, "Roll_no" : 105, "name" : "eee", "phone" : 415787784, "marks" : 75, "address" : "prqs", "year" : 2021, "town" : "Ernakulam" }
```

```
> db.student.updateMany({ },{ $unset: { address: "" } })

{ "acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 4 }

> db.student.find()

{ "_id" : 1, "Roll_no" : 101, "name" : "aaa", "phone" : 576524687, "marks" : 95, "year" : 2021, "town" : "Thrissur" }

{ "_id" : 3, "Roll_no" : 103, "name" : "ccc", "phone" : 575127587, "marks" : 75, "year" : 2021, "town" : "Kannur" }

{ "_id" : 4, "Roll_no" : 104, "name" : "ddd", "phone" : 62827784, "marks" : 99, "year" : 2021, "town" : "Thrissur" }

{ "_id" : 5, "Roll_no" : 105, "name" : "eee", "phone" : 415787784, "marks" : 75, "year" : 2021, "town" : "Ernakulam" }


> db.student.aggregate([{$addFields:{"apartment_no":"a23","street":"xyz","pin":680125}}])

{ "_id" : 1, "Roll_no" : 101, "name" : "aaa", "phone" : 576524687, "marks" : 95, "year" : 2021, "town" : "Thrissur", "apartment_no" : "a23", "street" : "xyz", "pin" : 680125 }

{ "_id" : 3, "Roll_no" : 103, "name" : "ccc", "phone" : 575127587, "marks" : 75, "year" : 2021, "town" : "Kannur", "apartment_no" : "a23", "street" : "xyz", "pin" : 680125 }

{ "_id" : 4, "Roll_no" : 104, "name" : "ddd", "phone" : 62827784, "marks" : 99, "year" : 2021, "town" : "Thrissur", "apartment_no" : "a23", "street" : "xyz", "pin" : 680125 }

{ "_id" : 5, "Roll_no" : 105, "name" : "eee", "phone" : 415787784, "marks" : 75, "year" : 2021, "town" : "Ernakulam", "apartment_no" : "a23", "street" : "xyz", "pin" : 680125 }


> db.student.update({_id:3},{$set :{"apartment_no":"b21","street":"hij","pin":683014}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.student.update({_id:4},{$set :{"apartment_no":"c72","street":"stu","pin":263014}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.student.update({_id:5},{$set :{"apartment_no":"n48","street":"klm","pin":673214}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.student.update({_id:1},{$set :{"apartment_no":"a23","street":"xyz","pin":680214}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.student.find()
```

{ "_id" : 1, "Roll_no" : 101, "name" : "aaa", "phone" : 576524687, "marks" : 95, "year" : 2021, "town" : "Thrissur", "apartment_no" : "a23", "pin" : 680214, "street" : "xyz" }

{ "_id" : 3, "Roll_no" : 103, "name" : "ccc", "phone" : 575127587, "marks" : 75, "year" : 2021, "town" : "Kannur", "apartment_no" : "b21", "pin" : 683014, "street" : "hij" }

{ "_id" : 4, "Roll_no" : 104, "name" : "ddd", "phone" : 62827784, "marks" : 99, "year" : 2021, "town" : "Thrissur", "apartment_no" : "c72", "pin" : 263014, "street" : "stu" }

{ "_id" : 5, "Roll_no" : 105, "name" : "eee", "phone" : 415787784, "marks" : 75, "year" : 2021, "town" : "Ernakulam", "apartment_no" : "n48", "pin" : 673214, "street" : "klm" }

> db.student.find({"name":"ccc"},{apartment_no:1, street:1,city:1,pin:1,_id:0})

{ "apartment_no" : "b21", "pin" : 683014, "street" : "hij" }

MICRO PROJECT

Aim: Using PHP and MySQL, develop a program to accept book information viz. Accession number, title, authors, edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings.

Source code:

```
<html>
<head>
<title>book</title>
</head>
<body align="center"><u><h1>Book Information System</h1></u><br>
<a href="add.html">Add Book</a><br>
<a href="search.html">Search Book</a><br>
</body> </html>
```

add.html

```
<html>
<head>
<title>addbook</title>
</head>
<body>
<center>
<h1>Enter Book Details</h1><br>
<form name="form1" action="add.php" method="POST">
BookNo:<input type="text" name="num"><br><br>
BookName:<input type="text" name="name"><br><br>
Author:<input type="text" name="auo"><br><br>
Publisher:<input type="text" name="pub"><br><br>
Edition:<input type="text" name="edi"><br><br>
<input type="submit" name="Submit">
<input type="reset" name="Reset">
</form>
</center>
```

```
</body> </html>
```

add.php

```
<?php
$num=$_POST['num'];
$name=$_POST['name'];
$author=$_POST['auo'];
$publish=$_POST['pub'];
$edi=$_POST['edi'];
$con=new mysqli("localhost","fisat","fisat","fisatdb"); if($con==false)
{ echo"Failed to connect";
} else
{ echo"connected";
}
$sql="INSERT INTO book32 VALUES($num,'$name','$author','$publish','$edi')"; if($con->query($sql))
{
        echo"<BR>"; echo"New
        row added";
} else {
echo"
ERROR
:could
not
execut
e
query"
;
}
$con->close(); ?>
```

search.html

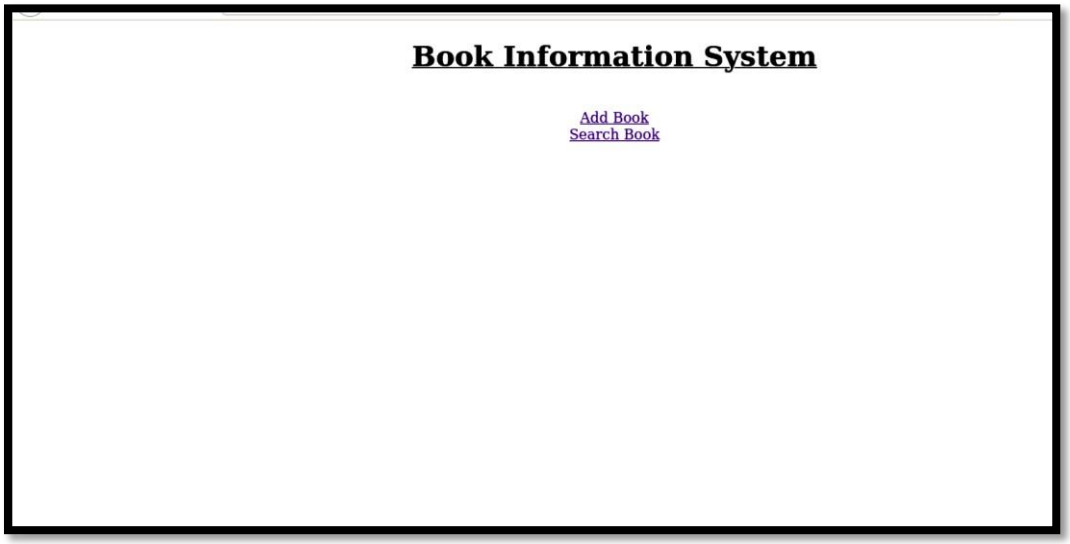
```
<html>
<head>
```

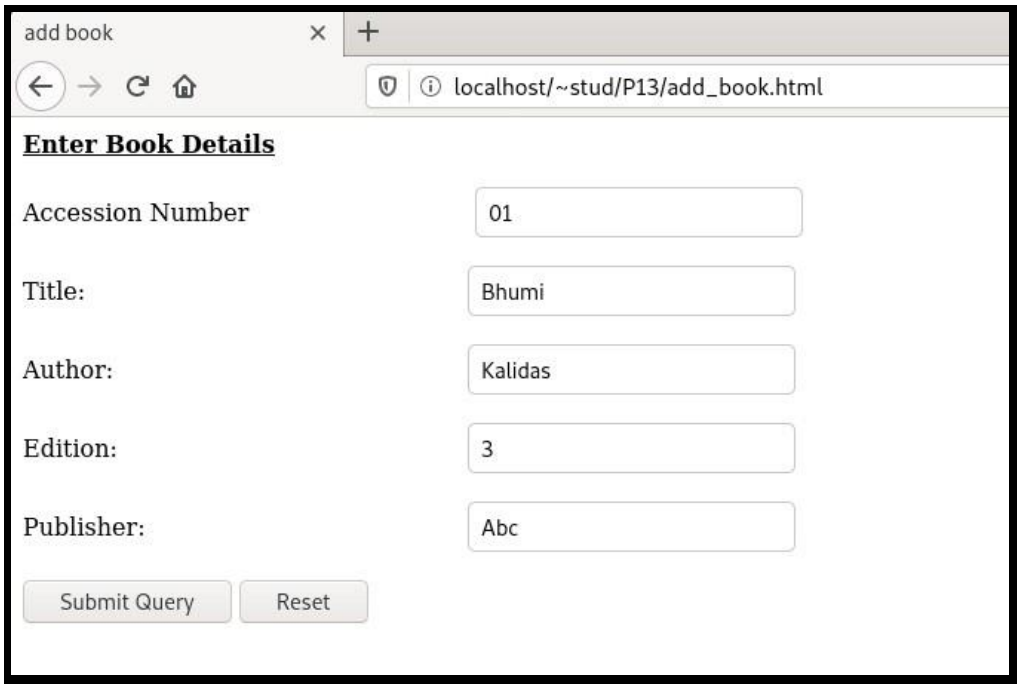
```
<title>search</title>
</head>
<body>
<center>
<form name="form2" action="search.php" method="POST">
<b><u><h1>SEARCH A BOOK</B><U/></h1>
Enter book title:<input type="text" name="title"><br><br>
<input type="submit" name="Submit">
</form>
</center>
</body> </html>
```

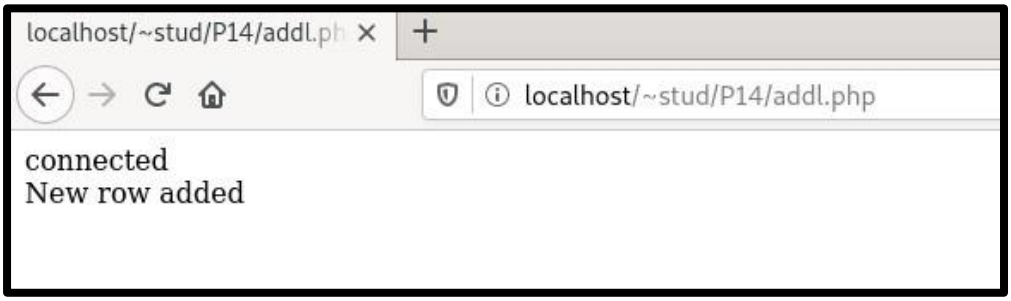
search.php

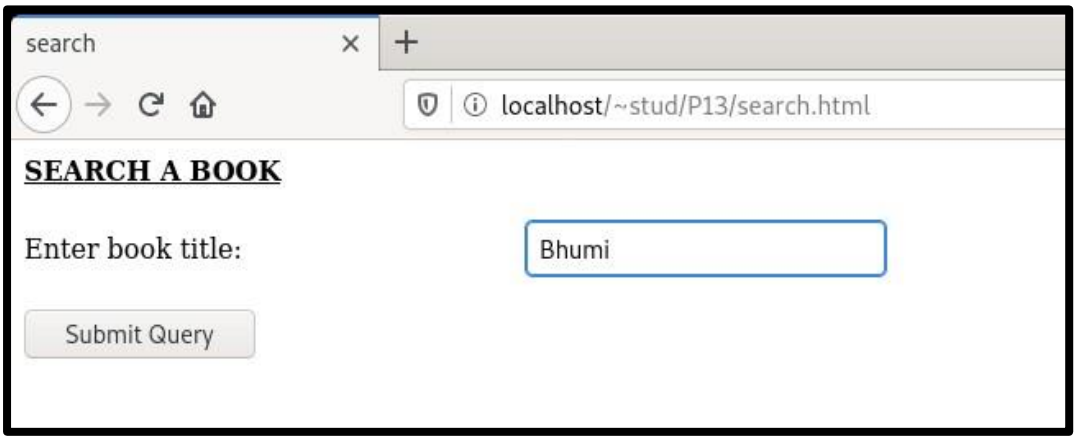
```
<?php
$title=$_POST['title'];
$con=new mysqli("localhost","fisat","fisat","fisatdb"); if($con==false)
{ echo"Failed to connect";
} else {
echo"c
onnec
ted"; }

$sql="SELECT * FROM book32 WHERE BookName='$title'"; if($result=$con->query($sql))
{ if($result->num_rows>0)
        { while($row=$result->fetch_array())
                { echo"\n".$row[0].":".$row[1].":".$row[2].":".$row[3].":".$row[4]."\n";
                }
                $result->close();
        } else
        { echo "\nCould not found the book"; } } else
{ echo "\nError:could not connect"; }
$con->close();
?>
```

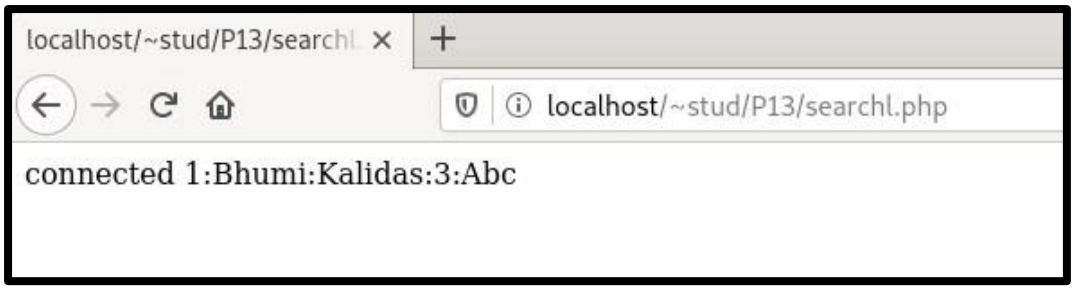
Output


Book Information System
Add Book
Search Book


add book
localhost/~stud/P13/add_book.html
Enter Book Details
Accession Number
01
Title:
Bhumi
Author:
Kalidas
Edition:
3
Publisher:
Abc
Submit Query
Reset


localhost/~stud/P14/addl.ph
localhost/~stud/P14/addl.php
connected
New row added


search
localhost/~stud/P13/search.html
SEARCH A BOOK
Enter book title:
Bhumi
Submit Query


localhost/~stud/P13/searchl
localhost/~stud/P13/searchl.php
connected 1:Bhumi:Kalidas:3:Abc